*Sinking Ships* is a game in which two players place a number of ships of various length on their own board, which is hidden from the other player. Players then take turns calling out one of the tiles on their opponent’s board. Their opponent then tells them whether the shot hit or missed any of their ships. Once one player has hit every tile that contains a ship on their opponent’s board, they win the game.

Using the following UML class diagram and class descriptions to help you, create a version of Sinking Ships. The game must have one human player, and one player controlled by the computer (the computer player can be made to target random tiles). The player should be able to choose how many rows (between 10 and 26) and how many columns (between 10 and 26) the game boards should have. Players cannot shoot tiles that they have already shot, and before the human player takes their shot, they should have the option to look at their own board (showing the location of their ships and the tiles that the computer player has taken shots at and whether those shots hit or missed) or the opponent’s board (showing the tiles that the human player has taken shots at, and whether those shots hit or missed, but not showing the location of the opponent’s ships). Player input should be given as a number, to indicate the column of their chosen location, and a letter, to indicate the row (first row is A, second row is B, etc.). You may use the Sinking Ships skeleton code to help you.

A diagram of a computer game

Description automatically generated

**Board**

|  |  |
| --- | --- |
| **Attribute/Method** | **Description** |
| columns | Specifies the number of columns on the game board. |
| rows | Specifies the number of rows on the game board. |
| board | Keeps track of the ship locations and shot locations on the game board. |
| playerName | Specifies the number of the player that the board belongs to. |
| Board() | Creates a new Board object. |
| display() | Displays the current state of the board, only showing ship locations if the player is looking at their own board. |
| getWidth() | Accessor for columns. |
| getHeight() | Accessor for rows. |
| takeShot() | Takes a shot at the given location on the board. |
| placeShip() | Asks the player to pick a location on the board and an orientation (vertical or horizontal) for a ship of a given length until a valid location and orientation are given that will fit the ship on the board, and then adds the ship to the board as specified. |
| checkWinner() | Checks whether all of the ships on the board have been sunk. |

**Player**

|  |  |
| --- | --- |
| **Attribute/Method** | **Description** |
| playerNumber | Specifies this player’s number. |
| playerBoard | Specifies this player’s board. |
| Player() | Creates a new Player object. |
| getNumber() | Accessor for playerNumber. |
| getBoard() | Accessor for playerBoard. |

**HumanPlayer**

|  |  |
| --- | --- |
| **Attribute/Method** | **Description** |
| placeShips() | Places all of this player’s ships onto their board, displaying messages to tell the player which ship they are placing and to confirm that they have successfully placed each ship. |
| takeShot() | Gets a valid location on a board and takes a shot at it, displaying a message and giving the player another chance to take a shot if they select an invalid target. |
| getColumn() | Gets a valid column on a board from player input, displaying a message and giving the player another chance to select a column if they select an invalid column. |
| getRow() | Gets a valid row on a board from player input, displaying a message and giving the player another chance to select a row if they select an invalid row. |

**ComputerPlayer**

|  |  |
| --- | --- |
| **Attribute/Method** | **Description** |
| placeShips() | Places all of this player’s ships onto their board, displaying one message to say that the computer is placing its ships and another message to say that all of the computer’s ships have been placed. |
| takeShot() | Gets a valid location on a board and takes a shot at it. |
| getColumn() | Gets a valid column on a board. |
| getRow() | Gets a valid row on a board. |